REMARKS

(A) Claim Rejections – 35 USC § 102:

In re section 2, the examiner noted that:

> "Claims 1-16 are rejected under 35 USC 102(e) as being anticipated by Pugh et al. (US 6,801,052)."

In the rejected claims, there are 3 independent claims, namely claims 1, 16 and 19. The three independent claims are listed for reference.

1. (Previously presented) An internal stage of a programmable look up table (LUT) circuit for an integrated circuit, comprising:
one or more secondary inputs; and
one or more configurable logic states; and
two or more LUT values; and
a programmable means to select a LUT value from said secondary input or said configurable logic state.

10. (Previously presented) A sub-circuit of a programmable look up table macro circuit for an integrated circuit, comprising:
M primary inputs, wherein M is an integer value greater than or equal to one, and each said M inputs received in true and compliment logic levels; and
2^M secondary inputs; and
2^M configurable logic states, each said state comprising a logic zero and a logic one; and
2^M LUT values; and
a programmable means to select each of said LUT values from a secondary input or a configurable logic state.

Applicant respectfully traverses the rejection. For anticipation under 35 U.S.C. § 102(a), Pugh et al. US 6,801,052 must teach every aspect of the claimed invention either explicitly or impliedly. Any feature not directly taught must be inherently present.

> "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference. "Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). "The identical invention must be shown in as complete detail as is contained in the ... claim." Richardson v. Suzuki Motor Co., 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim, but this is not an ipsissimis verbis test, i.e., identity of terminology is not required. In re Bond, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990)."

The applicant has disclosed a LUT circuit wherein the LUT can be portioned to pack more logic. The applicant makes use of configurable LUT values to create this new feature. In contrast Pugh has disclosed a method to combine 2 LUT circuits to map Boolean logic. Pugh makes no mention of LUT values. His method relies on combining two LUT outputs in AND, OR, XOR logic modules. The Applicant will show that Pugh does not demonstrate a plurality of claim elements in each of the independent claims, and dependent claims.

On page 28, lines 9-15 of the current application, the applicant states the following definitions for LUT circuits:

> "The term K-LUT refers to a look up table comprising K inputs. Such a LUT comprises 2^K LUT values, and at least one output. For a given combination of K-input values, a LUT value is received at said at least one LUT output. The term LUT tree and LUT cone refers to construction of a LUT, where there is a gradual decrease in the number of LUTs in each stage. A first of the K-inputs is common to all the LUTs in a first stage, a second of the K-inputs is common to all the LUTs in a second stage and so on until the last LUT stage is reached in a hard wired K-LUT tree."

Programmable LUTs are further described by the applicant on page 8, line 12 thru page 11 line 14. Specifically, on page 11, line 8, the applicant states:

> "A K-LUT cone or a K-LUT tree has K-input variables, K-stages and 2^K LUT values to realize a K-input function. Each stage has one common input variable. $2^{(K-1)}$ outputs from first stage feed as LUT values into second stage. Consecutive LUT value reduction continues until the last stage, when only 2 LUT values feed the last stage, and one LUT output is obtained."

As per applicant's definition, a LUT (Look-Up Table) circuit has a first stage, one or more internal stages and a last stage. Each LUT stage receives a common input variable (which is the control signal A shown in Fig-3A). The LUT values are described as the input values to each stage. Each stage input Looks-Up the LUT value based on the input state and passes it to the stage output. When the input state is one, it picks a first LUT value. When the input state is zero it picks a second LUT value. The very first stage has LUT values shown as configurable logic states in elements 303, 304 in Fig-3A. Each LUT value can be configured to a logic state one and a logic state zero. In Fig-3C the 2LUT is shown to have four LUT values, in Fig-3D the 3LUT has eight LUT values, and in Fig-3E the 4LUT has sixteen LUT values, and in Fig-3F the 5LUT has thirty-two LUT values at the first stage. In every case the last stage comprises only two LUT values. In Fig-3A, F is the LUT output and A is the LUT input signal. The value of the

LUT output F can alternate between the LUT values stored in 303 and 304 based on the logic signal state of input A.

With reference to Pugh Fig-1A, a first and a second 4LUT as defined by the applicant are shown in blocks 20 and 30 respectively. Each of the 4LUTs 20 & 30 has four inputs shown as 10, 11, 12 & 13. The LUT values are stored in memory bits inside of block 20 and 30 as stated in Col. 2, line 55 - Col. 3, line 4. These memory bit LUT values are not shown in Fig-1A. In the text Pugh shows 2^K LUT values (same as the applicant) for the first stage of the 4, 5 & 8 input LUT examples. Pugh does not show the internal construction of 4LUT 20. For any given combination of inputs 10-13, one memory bit LUT value out of 16 available LUT values is received at the 4LUT output (the node coupled to block 21, 22 & 41 which is not labeled) of the 4LUT 20 circuit.

<u>In re section 3 the Examiner noted that</u>:

> "With respect to claims 1, 2, 7 and 10, figure 1A of Pugh discloses an apparatus (10, examiner considers item 10 as a larger LUT circuit having other elements in it), comprising:
>
> one or more secondary inputs (other functions);
> one or more configurable logic states (output of LUT 20);
> two or more LUT values (LUT 20 has multiple values); and
> a programmable means (21, 22) to select a LUT value (value at X or Y) from a said secondary input or a said configurable logic state."

The values at X or Y do not constitute a Look-Up Table value as defined by the applicant. They represent the outputs of the larger LUT circuit 10 as identified by the examiner. The programmable means of MUX 21 allows selecting one of 3 inputs to MUX 21 to couple to MUX 21 output. MUX 21 output cannot Look-Up different MUX 21 input values based on an input signal to larger LUT 10. The 3 inputs to MUX 21 are: other functions, output of LUT 20 and output of AND 41. This is further disclosed by Pugh on Col. 3, line 30 as:

> "Besides an input terminal connected to the output terminal of the LUT 20, the multiplexer 21 has a second input terminal connected to an output terminal of the AND gate 41 and a third input terminal connected to other logic function circuit."

(i) In Fig-1A, Pugh does not demonstrate: <u>An internal stage of a programmable look up table (LUT) circuit</u>. This is a first reason to traverse the rejection. In the larger LUT circuit 10, there is a LUT 20, an AND gate 41, four MUXs 21, 22, 25 & 26 and two Registers 23 & 24. MUXs 21

& 22 are not used as internal stages of the larger LUT 10 as they do not receive a LUT input signal (for example a 5^{th} input similar to 10-13). This is specifically stated by Pugh in Col. 3, line 24 as:

> "In passing, it should be noted that control lines to the multiplexers in Fig-1A (and following drawings) are not shown. It is understood that control signals which govern the selective operation of the multiplexers are set by configuration bits of the FPGA."

In other words, Pugh states that by setting the FPGA configuration bits, one of 3 inputs to MUX 21 can be selected to the MUX output (which acts as the input to the register 23). In the larger LUT 10, there is only one 4LUT 20 comprising 4 LUT stages, and no other LUT stage to create a larger LUT function. This fact is further obviated by Fig-1B, wherein Pugh discloses that the dotted line 40 comprises AND 41, OR 42 and XOR 43 gated by MUX 44. The user can select one of AND, OR, XOR elements to combine outputs of LUT 20 and LUT 30. A fifth LUT stage would have eliminated these logic components as a 5^{th} LUT stage can combine the outputs of the two 4LUTs 20 & 30 into any given logic combination. This method of LUT cone construction is illustrated by the applicant in Figs. 3B-3F and described on page 10, starting at lines 16. A further illustration of combining a LUT 1501 output and LUT 1502 output by a LUT stage 1503 is shown in Fig-15 (page 50, lines 4-12).

(ii) In Fig-1A, Pugh does not demonstrate: <u>a programmable means (21, 22) to select a LUT value (values at the inputs to the MUX) from a said secondary input or a said configurable logic state.</u> This is a second reason to traverse the rejection. As discussed in the previous section (i), the MUX 21 is not a LUT (Look-Up Table) stage. Hence there is no meaning to the MUX 21 having LUT values as it cannot Look-Up alternative LUT values based on the polarity of the MUX input signal. However, the MUX 21 does receive 3 secondary inputs: other functions, output of LUT 20 and output of AND 41. The programmable means in 21 does allow one of these inputs to couple to the output; while not providing different Look-Up values to the output. As the examiner correctly identified, the 4LUT 20 has 4 inputs and 16 LUT values – each LUT value implemented with memory as a configurable logic state. Four inputs 10-13 couple one of these LUT values to the 4LUT output. There are no secondary inputs inside the 4LUT 20. There is no provision inside 4LUT 20 to select either a configurable logic state or a secondary input as a LUT value for the first stage, or the intermediate stages or the last stage of 4LUT 20.

(iii) With respect to claim 10, in Fig-1A, Pugh does not demonstrate: 2^M secondary inputs; and 2^M configurable logic states, each said state comprising a logic zero and a logic one; and 2^M LUT values. This is a third reason to traverse the rejection. With reference to the larger LUT 10 in Fig-1A, there are 4 inputs and $2^4 = 16$ configurable logic states implemented as memory elements inside 4LUT 20. These elements are hard-wired as LUT values to the 4LUT – similar to Fig-3E shown by the applicant. There is no matching number of 16 secondary inputs, or a choice to program and select either a logic state or a secondary input as a LUT value.

Furthermore, Pugh fails to identify a motivation for the LUT circuit to be constructed such that an internal stage of the LUT can be configured as claimed by the applicant. In Fig-1A, let the output of 4LUT 20 be F and output of 4LUT 30 be G. (Please note that the user might need F for a first logic function and F' (not F) for a second logic function). Pugh provides 3 separate logic elements (AND 41, OR 42, XOR 43 in Fig-1B) to every 4LUT to combine F and G. The results are one of the following 3 Boolean functions: FG, F+G, FG+F'G'. Clearly there are many other possibilities: F'G, FG', F'+G, F+G', (FG)', (F+G)', F+F'G, etc. Should the hardware designer then provide all these features inside block 40 in Fig-1A? There is no "apriori" reason that only those three selected logic features by Pugh are desired. So any reader of this invention would have no motivation to create a plurality of pre-assigned logic functions at every single 4LUT output location. The overhead would be simply enormous. Furthermore, the combined output function becomes an 8-input function. Hence, Pugh uses an 8 input LUT function on Col. 2, line 58 as a motivating factor. No commercial logic fracture tool create 8-input LUT functions as the base line. Commercial tools fracture logic into 2-input equivalent logic gate clusters (say 2-input NAND gates). This is the basic reason why ASIC designers use the terminology "Equivalent System Gate Density" to describe complexity of designs. A system gate is a 2-input gate, typically a 2-input NAND gate. Any 2-input gate fits in one basic 2-input LUT. To map logic into FPGA devices, these 2-input NAND gates are re-assembled into larger 3-input and 4-input functions. At the end, one is left with 2LUT, 3LUT and 4LUT pieces. There is no motivation from the tools side to create 8-input Boolean functions as the basic building block. All the pieces are then dumped into 4LUT blocks.

The applicant has provided a reference (Ahmed Ref-3) for 20 bench mark designs ported into FPGAs as listed on page 11, line 14, and summarized in Fig-4. Details of the logic blocks for these designs comprising 4LUTs is given in Tab-3 of the reference (a copy of the reference was submitted by the applicant with the IDS disclosure statement). These 4LUTs are 2LUTs +3LUTs +4LUTs from the previous reconstruction efforts. The applicant has disclosed multiple embodiments as described in page 46, line 12 – page 56, line 17 to illustrate the advantages with the applicant's disclosure.

(iv) With respect to claim 2, in Fig-1A Pugh does not demonstrate: <u>a plurality of LUT values, each of said LUT values selected from a secondary input or a configurable logic state</u>. This is a third reason to traverse the rejection. As discussed in (ii) the MUX 21 is not a LUT stage. Hence there is no means for the MUX 21 to comprise a plurality of LUT values. MUX 21 only receives 3 secondary inputs, and only one is selected by the configuration bit.

(v) With respect to claim 7, in Fig-1A Pugh does not demonstrate: <u>a means of providing said secondary input as an output when said configurable logic state is selected as a LUT value</u>. This is a third reason to traverse the rejection. In Fig-1A, there is no configurable logic states coupled to MUX 21. It receives the 3 secondary inputs to MUX 21; one of which can be programmed to couple to the larger LUT 10 output X. The hard-wired configurable logic states inside 4LUT 20 has no impact on the MUX 21 programmable choice.

(vi) In re section 4, with respect to claim 3 & claim 11, in Fig-1A, Pugh does not demonstrate: <u>a memory bit programs said selection between secondary input and configurable logic state</u>. This is a third reason to traverse the rejection. The memory bits in MUX 21 programs one input from the three available secondary inputs. There is no configurable logic state feeding the MUX 21. The configurable logic states are located inside 4LUT 20. The inputs 10-13 (not memory bits) select which LUT value couple to 4LUT 20 output.

(vii) In re section 5, with respect to claim 4, in Fig-1A, Pugh does not demonstrate: <u>2^M LUT values, each said LUT values comprising a configurable logic state or a secondary input, wherein any given combination of said M primary input signal levels couples one of said LUT values to</u>

said LUT output. This is a third reason to traverse the rejection. The 2^M LUT values are inside 4LUT 20. There are no 2^M secondary inputs inside of 4LUT 20. There are 4 LUT inputs 10-13. For any given combination of these 4 LUT inputs 10-13, the larger LUT 10 output X could be anything. For example, if MUX 21 was programmed to select the "other function" as the secondary input, then regardless of inputs 10-13, the 4LUT 20 values do not couple to larger LUT 10 output X.

(viii) In re section 6, with respect to claim 6, in Fig-1A, Pugh does not demonstrate: a secondary input comprised one of a register output and a memory output. This is a third reason to traverse the rejection. In Col. 3, line 33 Pugh discusses other logic functions that make up other functions. In Fig-1B he shows logic blocks for element 40. Pugh does not describe memory or register outputs as possible inputs to MUX 21.

(ix) In re section 7, with respect to claims 8 and 12, in Fig-1A, Pugh does show a K-LUT output as a secondary input. However, the independent claim elements were not demonstrated.

(x) In re section 8, with respect to claims 9 and 14, in Fig-1A, Pugh only disclose SRAM memory elements. The independent claim elements were not demonstrated.

(xi) In re section 9, with respect to claims 15 and 16, in Fig-1A, Pugh does not demonstrate: wherein programmable selection of one or more of said secondary inputs as LUT values further comprises implementing a (K+M) input LUT function (and) wherein programmable selection of 2^M configurable logic states as LUT values further comprises implementing a M-LUT function decoupled from 2^M other K-LUT functions. This is a third reason to traverse the rejection. Pugh does not have the means to construct an (M+N) LUT function. For example, in Fig-1A, Pugh has 4LUT 20 (M=4) and 4LUT 30 (N=4), but he cannot construct an 8LUT (M+N=8). He can only construct 3 specific AND, OR, XOR combinations of the two 4LUT outputs – which is far shorter than a full 8LUT logic capability. Similarly Pugh cannot create an M-LUT function decoupled from 2^M other K-LUT functions either. He has access to the outputs of completely decoupled 4LUT 20 and 4LUT 30.

The applicant has presented either 2 or 3 claim elements in each of the claims not demonstrated by Pugh. One with ordinary skill would not have the rationale to modify the Pugh invention to construct a LUT macro-cell wherein the LUT can be partitioned into smaller pieces. Pugh discloses a technique to combine 2 4LUT outputs with pre-assigned logic blocks. The many omitted possibilities to combine the two outputs coupled with no apriori knowledge of which functions to keep would prevent a reader endorsing the Pugh LUT function. Therefore the internal LUT stage with selectable LUT values is not expressly or impliedly stated in the listed prior art, and a *prima facie* case of obviousness has not been met; it would not be obvious to one with ordinary skill to arrive at the Applicants disclosure stated in the independent Claim 1 and Claim 10 and those dependent thereupon. Applicant's rejected claims 1-16 would have not been obvious in view of Pugh. Withdrawal of the rejection of Claims 1-16 is respectfully requested.

<u>(B) Claim Rejections – 35 USC § 103:</u>

<u>In section 10, the examiner noted that:</u>

> "Claims 5 and 13 are rejected under 35 USC 103(a) as being unpatentable over Pugh et al (US 6,801,052) in view of Sugibayashi et al. (US 6,515,551)."

Applicant respectfully traverses the rejection. In both of the rejected dependent claims, Claim-5 & Claim-13, the applicant has claimed the following: <u>the circuit of independent claim, further comprising a thin film transistor</u>. In previous section (A) the applicant has discussed multiple elements in the claim language that is not demonstrated by Pugh. While Sugibayashi discloses a thin-film transistor, he does not disclose a technique to construct partitionable LUT macro-cells for an FPGA. Specifically, Sugibayashi discloses using a TFT device for a transfer gate, and does not disclose or discuss LUT circuits. Every claim element in the independent claims 1 & 10 are explicitly or impliedly absent in the Sugibayashi disclosure. Hence, all of the missing elements listed under sec A (with reference to Pugh) are also absent in Sugibayashi. Therefore claims 5 & 13 would not be obvious to one with ordinary skill in view of Pugh, and further in view of Sugibayashi. Withdrawal of the Claim 5 & Claim 13 rejection is respectfully requested.

CONCLUSION

The applicant believes that the above submission is fully responsive to the office action.

If for any reason the Examiner believes that a telephone conference would in any way expedite this matter, the Examiner is invited to telephone the Inventor Mr. Madurawe at (408) 737-8868 or on his cell phone at (408) 431-5367.

Respectfully submitted,

Ramindall Madurawe

Raminda Madurawe